

SimLack: Simulation-based Optimization and Scheduling of Generic Powder Coating Lines

Lukas Hollenstein^{1*}, Adrian Lötscher¹, Fabian Luccarini²

¹ZHAW Zurich University of Applied Sciences, IAS Institute of Applied Simulation, Grüental, Postfach, 8820 Wädenswil, Switzerland; *lukas.hollenstein@zhaw.ch

²e. Luterbach AG, Mülacher 10, 6024 Hildisrieden, Switzerland

SNE 29(3), 2019, 127 - 132, DOI: 10.11128/sne.29.tn.10483
Received: June 10, 2019 (Selected ASIM SST Hamburg 2018 Postconf. Publ.), Accepted: July 22, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Powder coating and paint-spray lines are often complex production plants because of many dynamical dependencies, limited buffer space and sequence dependent changeover times. We have developed a generic simulation and optimization platform that enables the engineers to design more performant and energy efficient facilities and the production planners to increase productivity through simulation-based optimization. The simulation environment builds on a generic modelling library that captures all variations of such facilities. 'Executable' models are generated automatically from annotated CAD layouts. As a result, the system smoothly integrates with the engineering process. Once the facility is in use, the fully specified virtual plant is used for simulation-based scheduling, employing a combination of a generic priority-based heuristic and a variant of simulated annealing. We discuss how these two aspects of the system render it an important innovation for the painting line industry and show first results from the scheduling system.

Introduction

Powder coating and paint-spray facilities are among the most important energy consumers in the industry, mainly due to their cure ovens staying in operation all day long. To keep operating costs and environmental impact down, machinery manufacturers, like the company e. Luterbach AG, are aiming to optimize the facility layouts and dimensions to their customers' needs.

Once a powder coating line is installed and productive, the operating company intends to plan their production schedule in order to maximize productivity and minimize operation cost. Such facilities normally rely on a closed loop material handling system that transports the parts to be treated on hangers along chains. Failing to feed the system with an efficient job sequence can lead to a significant drop in throughput or even to deadlocks.

Since a few years, Luterbach is already at the forefront of designing energy efficient facilities through simulating the energy and heat budget by means of pinch analysis [1]. The next step is to quantitatively optimize the design and the usage strategies of such systems with respect to the material flow and production performance. Even though manufacturing resource planning (MRP II) for generic production facilities may exist on the market the specific industry sector (powder coating and paint-spraying) has not embraced such solutions. Plant operators are often not employing enterprise resource planning (ERP) software throughout the production process or, if they do, struggle to enable it with sophisticated scheduling abilities that go beyond simple heuristics based on the planner's experience. In addition, when designing new facilities and retrofits, it appears that most machinery manufacturers still rely on static analysis and experience to plan the facility layouts. These approaches are bound to fail as facilities grow larger, more flexible and complex.

The scheduling problem for generic powder coating and paint-spray facilities can in principle be seen as hybrid flow shop scheduling with unrelated parallel machines [2, 3, 4]. Unfortunately, the rules that govern the dynamics of the facilities can be highly specific and complex, as can be the routes of parts through the system and the timing involved.

As a result, the scheduling problem for such manufacturing systems is not straightforward to formulate in closed form in order to apply mathematical programming techniques or well-known heuristics. Therefore, a simulation-based approach is more promising [5, 6].

Apart from scheduling, there are two more reasons for a simulation approach in the context of plant engineering: first, the facility design phase highly profits from a generic simulation framework than can be used to rapidly build models and compare different scenarios and configurations. Secondly, a dynamic visualization of the system helps to understand it, to pinpoint bottlenecks and, importantly so, supports sales with a quantitative proof of concept. Wilson & Zettle report on an operative scheduling solution for powder coating lines that relies on existing simulation models developed in the planning phase of the facility [7]. However, it is unclear in which way the simulation is used in their heuristic scheduling algorithms and if the scheduling methodology is generically applicable to any facility configuration.

We extend this approach in two ways. Firstly, we specify a generic annotation scheme for computer aided design (CAD) layouts that enables automatic model generation for powder coating lines. Secondly, we discuss a scheduling methodology that combines a generic priority-based sorting heuristic with a variant of the simulated annealing meta-heuristic that improves the solution by means of many simulation runs. We have implemented these concepts as an integrated platform for simulation-based optimization and scheduling of powder coating and paint-spray facilities. The simulation environment is called *Sim-Lack*, derived from *Simulation* and *Lackieranlagen*, the German word for paint shop.

1 Automatic Model Generation from CAD Layouts

When planning new powder coating and paint-spray facilities or designing a retrofit, machinery manufacturers aim at optimizing the facility layout and number and dimensions of building blocks in order to minimize the energy consumption and maximize the total throughput. Modelling and simulations enable prototyping, iterative refinement and quantitative analysis of designs.

However, firstly, building and analyzing simulations is time consuming and, secondly, the engineers are usually no simulation experts.

To overcome this issue, we have developed an integrated simulation environment that consists of a generic modelling library for painting lines with an associated domain specific language based on annotated CAD drawings to generate simulation models automatically. A discrete-event simulation engine runs the models while showing the model state as a schematic visualization auto-derived from the original CAD drawing.

1.1 The generic model

Painting line facilities all follow similar principles and consist of the same building blocks, but vary in their details. A hanging conveyor system of some kind (often a *Power and Free* chain) is carrying the parts to be treated in hangers through a series of processing steps including pre-treatment, powder application or paint-spraying, curing, and cooling. Other intermediate steps as well as buffering and inner closed loops can be present, which complicate the system. Due to space and cost restrictions, the lines typically cannot be kept lean such that material and color-dependent changeovers have to be taken into account and dynamic dependencies appear at certain points in the facility.

The generic discrete-event model describes each job as a number of parts on a hanger. The hangers are the entities that are moved through the system. The system is defined by the *facility configuration* that consists of

- the *positions* at which the hangers can reside,
- the *segments* of the hanging conveyor network connecting the positions,
- the *decision points* where two segments join or separate, and
- the *processes* in which the hangers can change state, representing either a loading point (source), a generic process (pretreatment, powder application, curing, cooling, etc.), or a discharging point (sink).

Each position belongs to exactly one process. Each segment consists of a sequence of positions. The segments must form a closed loop globally.

Next to the facility configuration, the model is specified by the following additional data:

- the sequence of processes a job has to pass through (incl. processing time or minimal/maximal retention time),
- specialized prioritization rules per decision point (optional), and
- sequence dependent changeover times per process (optional),
- availability patterns of processes to represent shifts and planned outages (optional).

The entities (hangers) are moved from one position to the next along the segments and cannot overtake other entities. Positions can have capacity larger than one to allow for quickly adding buffer space in experiments. The routing of entities through the system is derived from the job's sequence of processes and the prioritization rules at decision points that can depend on the job attributes and the state of processes.

The model is generic in the sense that it is agnostic about the actual meaning of the processes and even the attributes of the jobs that influence the processing time, the changeover times and the prioritization rules in decision points. The semantic behind the processes and the job attributes is only in the data that is provided to generate the simulation model. This means the model represents very generically any kind of closed-loop flow shop system. It is therefore widely applicable, not only to the partnering company and not even only to painting lines.

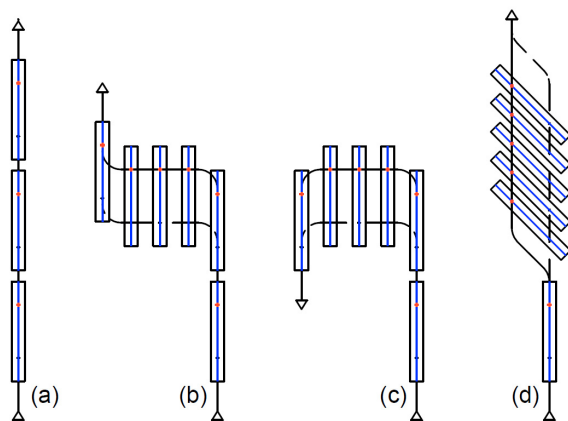


Figure 1: Example snippets from a CAD layout. Segment (a) has three positions in longitudinal direction; (b) is a transversal buffer where hangers exit in the same direction as they enter, i.e. no change of orientation; (c) is a transversal buffer with change of direction; and (d) is a diagonal buffer without change of direction.

1.2 The CAD-based domain specific language

In order to generate executable instances of the generic simulation model the complete facility configuration incl. its data needs to be provided. Machinery engineers use CAD drawings when designing new plant layouts or investigating changes to existing facilities. To integrate the simulation library seamlessly into the workflow of the engineers we devised an annotation scheme for CAD layouts. The annotated CAD layout is imported into the simulation environment and turned into a specific simulation model automatically. The data needed to fully specify the simulation experiment (jobs, prioritization rules, changeover times, and availabilities) is not provided through the CAD-interface and needs to be imported separately. Here we focus on the CAD-based domain specific language, as it is the main innovation that enables automatic model generation.

CAD layouts of painting lines are complex and contain many elements, most of which do not provide information for generating the model. Fig. 1 shows snippets from a CAD layout exemplifying sequences of positions (i.e. segments). A drawing does not naturally contain semantic information, e.g. even though positions are always rectangular shapes it is not clear from the outset which process the positions belong to. Therefore, we require the elements to be placed in blocks (aka layers) that are named semantically and structured hierarchically according to the following scheme:

- **Block Processes:** contains all processes, e.g.
 - **Block LoadingPoint:** contains all its positions, e.g.
 - Block P001 (with rectangle)
 - Block P002 (...)
 - **Block Buffer1:** e.g.
 - Block P003 (...)
 - ...
 - Block P010 (...)
 - ...
- **Block Segments:** contains all segments, e.g.
 - **Block P000-P002_0** (with lines and arcs)
 - **Block P002-P010_1** (...)
 - ...

There are optional naming conventions for the processes such that the type of process can be recognized (loading point, generic process, or discharging point). The positions can be named arbitrarily but uniquely. In the example above we applied the optional convention to name the positions P001 ... P010.

In order for the importer to uniquely identify the starting and ending positions of segments, we require the following convention: the names of segments must be made up of the *last position of the preceding segment* and the *last position of the current segment*. For example the segment P000-P002_0 starts at the position P000, that is the end of the preceding segment, and ends at position P002. The importer algorithm identifies the intermediate positions along the segment geometrically.

Segments feature another complication, namely the possibility that the orientation of the hangers can change in the end of a transverse buffer, see Fig. 1. Because such situations are not straightforward to detect algorithmically we require the name of the segment to end either with _0 for “no change of orientation” or with _1 for “change of orientation.”

The combination of CAD drawing and annotation scheme (naming convention) can be seen as a domain specific language that enables data-driven model generation. Because the annotation with specifically named blocks is lightweight in its application, this approach integrates neatly into the workflow of the plant engineers. Other possibilities such as specifying the facility configuration in tables or using a graphical drag-and-drop editor would certainly lead to significant overhead in the design cycles. The specification is domain specific in the sense that it does rely on consistency relations that hold specifically in the powder coating domain, even though it may be applicable more generally to other closed-loop hybrid flow shop systems.

2 Simulation-based Optimization

Production planners of painting lines currently use experience-based heuristics and simple static estimates to put together production schedules given a number of orders.

Algorithmic assistance is highly asked for; however, acceptance of decision support systems strongly depends on the level of control and understanding the user can have.

With this requirement in mind, we have developed a hybrid scheduling scheme that builds on the user’s planning experience and knowledge of the facility and product characteristics. The method works in two phases that we describe briefly in the following.

2.1 Priority-based heuristic

- The user selects a number of *sorting priorities*. These are job attributes like material, paint color, due date, etc. and can be configured freely for each specific facility.
- All permutations of the sorting priorities are generated.
- Each permutation is then used to generate a job sequence by sorting hierarchically according to that sequence of priorities.
- For n selected attributes this procedure yields $n!$ priority-sorted job sequences.
- All sequences are evaluated using the simulation model of the facility, respecting all constraints like shift plans, changeover times, outages, etc.

The result is twofold: A good estimate of the optimal schedule (according to some objective function like the makespan) and insight into which job attributes have a strong impact on the performance of the job sequence. This insight helps to improve acceptance by the user and the user can explore and learn about the facility.

2.2 Simulated annealing for refinement

Starting from the best job sequence found by the priority-based heuristic, we employ a variant of *simulated annealing*. This class of stochastic meta-heuristics is well-known to be able to find globally near-optimal solutions with certainty and has proven convergence properties [6]. The basic algorithm works as follows:

- In each iteration a random solution in the neighborhood of the current solution is generated.
- The objective function f is evaluated and if the new solution is better than the current one it is accepted directly as the next step.

- If the new solution is not better, it is accepted all the same with a probability proportional to $e^{-\Delta f/\theta_i}$. Here Δf is the positive difference between the new and the current objective value. The parameter θ_i is a positive number that depends on the iteration i . It starts relatively large which means the probability of accepting poor solutions is large and therefore exploration of the solution space is made possible. With increasing number of iterations θ_i is decreasing such that poor solutions are only seldom accepted, exploration ceases and exploitation (local refinement) is emphasized.
- As soon as the globally best solution does not improve anymore for some number of iterations or the maximal number of iterations is reached the algorithm stops.

The difficulty in applying simulated annealing is to define an efficient solution neighborhood, i.e. a procedure for generating new solutions nearby a given solution. A simple choice would be to swap a random single pair of adjacent jobs in the sequence, *single adjacent interchange*. We compared the performance of a number of different neighborhoods and found that the scheduling problem at hand asks for more exploration than single interchanges. Therefore, we make the neighborhood depend on the iteration: the number of adjacent interchanges starts at the length of the job sequence, i.e. large exploration, and subsequently reduces to one with increasing number of iteration. A similar effect could be achieved with single adjacent interchange and θ_i reducing more slowly, but we found this variant to converge more quickly for a set of reference problems.

2.3 Preliminary results

Figure 2 compares random sampling of the solution space with the proposed probability-based heuristic and simulated annealing. We show the distribution of objective values from the evaluated job sequences. In this case, the objective function is the makespan that is minimized. The distributions is multi-modal (three clusters appear) which is due to the discontinuity of the objective function. Observe that random sampling is not able to find the best-performing solutions with makespan below 160 h. The priority-based heuristic generates a broad range of solutions incl. some that come close to the best ones found.

The simulated annealing algorithm, however, is able to improve on the sorting heuristic and spends most of its time near the optimum trying to improve on it locally. These are preliminary explorations and more systematic performance analyses and comparisons are left for future work.

3 Summary and Outlook

The integrated simulation and optimization environment *SimLack* essentially provides two important innovations to the industry sector of powder coating and paint-spray lines. Firstly, a generic modelling library and a domain specific language were developed to provide the possibility to generate simulation models automatically from CAD layouts. This approach integrates the simulation environment seamlessly into the workflow of the plant engineers. Secondly, a simulationbased scheduling methodology was devised, based on a combination of a generic sorting heuristic and a variant of the simulated annealing meta-heuristic.

After importing a CAD layout into *SimLack*, a graphical user interface eases tweaking the facility configuration, data import and editing, the specification of detailed prioritization rules to apply at decision points, setting up changeover criteria and durations, and shift plans. The integrated platform enables the plant engineers to set up, carry out and organize simulation experiments to compare different facility layouts and various parameter settings.

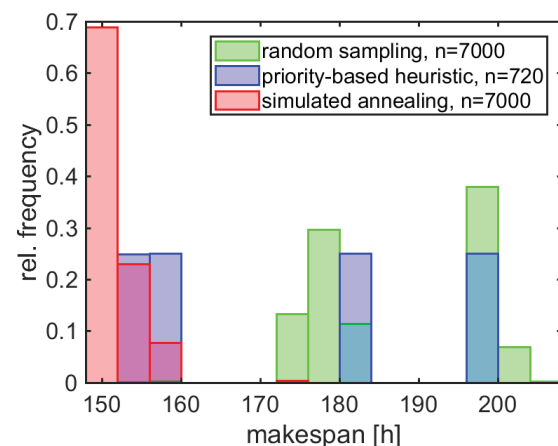


Figure 2: Comparison of different algorithms in terms of the distribution of the objective values (makespan) from evaluated job sequences in the course of the optimization runs.

A first use-case shows that e. Luterbach AG was able to acquire a new contract successfully through the ability to fine-tune the plant details and to convince the customer with quantitative estimates of its performance. In this case, the seamless integration of the *SimLack* platform proved crucial for its success, as the engineers went through several iterations to refine the layout to the point where requested performance could be achieved.

Preliminary performance tests of the scheduling system with small and medium sized problems show:

- Even though the sorting heuristic in the first stage can provide relatively good solutions the second stage can typically improve on it in a few thousand iterations (simulation runs), depending on problem size.
- Our simulated annealing variant with a neighborhood with decreasing number of adjacent interchanges is converging more efficiently than variants with a constant number of adjacent interchanges.
- Since simulations are run in parallel, a scheduling run for a large sequence of 790 jobs on a normal sized facility takes about 30 min (for 7000 simulations) on a quad-core laptop to run.
- These first results are encouraging to refine details of the *SimLack* system and its algorithms. Next to more in depth performance analysis, we are planning to add alternative algorithms that the user can choose from and compare against each other. Further, a challenging improvement is to add support for the plant design process by providing means of automatically optimize certain parameters and aspects of a facility configuration.

Acknowledgement

We thank Florian Eggenberger and Ramon Kramis from e. Luterbach AG and Ivo Berchtold and Othmar Winiger from wirth+co AG for their constructive collaboration and the two companies for their support. This work is supported by the Commission for Technology and Innovation (CTI) of the Swiss Confederation through grant number 17655.1 PFEN-ES.

References

- [1] Ke. A quantum leap. *IST International Surface Technology*. 2017; 10(1):16–21.
- [2] Pinedo ML. *Planning and Scheduling in Manufacturing and Services*. New York: Springer-Verlag, 2nd ed. 2009.
- [3] Gupta JND, Stafford EF. Flowshop scheduling research after five decades. *European Journal of Operational Research*. 2006;169(3):699–711.
- [4] Ribas I, Leisten R, Framiñan JM. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*. 2010; 37(8):1439–1454.
- [5] Kiran AS. Simulation and Scheduling. In: *Handbook of Simulation*, edited by Banks J, This chapter contains sections titled: Introduction, Definitions and Background, Scheduling Approaches, Simulation of Job Shops, Simulation-Based Scheduling: Application, Implementation Issues, Concluding Remarks, References, pp. 677–717. Wiley-Blackwell. 2007.
- [6] Gosavi A. *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Operations Research/Computer Science Interfaces Series. Springer US, 2nd ed. 2015.
- [7] Wilson H, Zettl R. Simulation and Optimization Based Flexible Job Scheduling of Powder Coating Lines. In: *Digital Product and Process Development Systems*, edited by Kovács GL, Kochan D, IFIP Advances in Information and Communication Technology. Springer, Berlin, Heidelberg. 2013; pp. 369–375.